

In-Vehicle Computing

Hardware Platforms for Mobile Applications



**LVC-2001
V1.2**

User's Manual

Release Date: 2022/08/04

Overview

Icon Descriptions

The icons are used in the manual to serve as an indication of interest topics or important messages. Below is a description of these icons:



NOTE: This check mark indicates that there is a note of interest and is something that you should pay special attention to while using the product.



WARNING: This exclamation point indicates that there is a caution or warning and it is something that could damage your property or product.

Online Resources

To obtain additional documentation resources and software updates for your system, please visit the [Lanner Download Center](#).

In addition to contacting your distributor or sales representative, you could submit a request to our [Lanner Technical Support](#) to fill in a support ticket to our technical support department.

Copyright and Trademarks

This document is copyrighted, © 2022. All rights are reserved. The original manufacturer reserves the right to make improvements to the products described in this manual at any time without notice.

No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of the original manufacturer. Information provided in this manual is intended to be accurate and reliable. However, the original manufacturer assumes no responsibility for its use, nor for any infringements upon the rights of third parties that may result from such use.

Acknowledgement

Intel®, Pentium and Celeron® are registered trademarks of Intel Corp.

Microsoft Windows and MS-DOS are registered trademarks of Microsoft Corp.

All other product names or trademarks are properties of their respective owners.

Compliances and Certification

CE Certification

This product has passed the CE test for environmental specifications. Test conditions for passing included the equipment being operated within an industrial enclosure. In order to protect the product from being damaged by ESD (Electrostatic Discharge) and EMI leakage, we strongly recommend the use of CE-compliant industrial enclosure products.

FCC Class A Certification

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

e Mark Certification

E13 - Luxembourg

Mechanical compliance

Vibration:

- General Vibration (operating): Refer to MIL-STD-810G, Method 514.6, Procedure I (Transportation), Category 4 – Common carrier (US highway truck vibration exposure)
- General Vibration (non-operating): Refer to MIL-STD-810G, Method 514.6, Procedure I (Transportation), Category 24 – General minimal integrity

Shock:

- Operating (Functional Test for Ground Equipment): Refer to MIL-STD-810G, Method 516.6, Procedure I, 40g, 11ms
- Non-Operating (Crash Hazard Shock Test for Ground Equipment): Refer to MIL-STD-810G, Method 516.6, Procedure V, 75g, 11ms

Electrical transient conduction along supply lines only
(12V/24V)

Revision History

0.1	2015/06/15	Preliminary
0.2	2015/06/30	Revised specifications: added vibration and wall mount drawings. Modified hardware setup
1.0	2016/04/28	Official release
1.1	2016/10/03	Added SIM card installation

Table of Contents

Chapter 1: Introduction	5
System Specifications	5
Package Contents	6
Ordering Information	6
Chapter 2: System Components	7
Mechanical Drawings	7
Block Diagram: The Main Board	10
Front Components	11
Rear Components	12
Chapter 3: Motherboard Information	13
Connectors and Jumpers	13
Add-on Card: LVK-MSDM	15
Connectors and Jumpers List	16
Jumper Settings	17
Connectors	17
Chapter 4: Hardware Setup	24
Preparing the Hardware Installation	24
Disk Drive Installation	24
Installing Memory Module	24
Installing Full-sized Mini-PCle Module	24
Installing mSATA Storage Device	25
Installing Half-sized Mini-PCle Module	25
Full-sized Mini-PCle Module Installation	25
Installing a SATA/mSATA Storage Device	25
Connecting Power	26
Installing SIM Cards	27
Chapter 5: The Flow Chart	28
Appendix A: Using the Ignition System Manager (ISM)	29
Appendix B: Digital Input/Output	30
Appendix C: Accessing the GPS Data	36
Appendix D: Programming System Watchdog Timer	38
Appendix E : Terms and Conditions	42

Chapter 1:

Introduction

Thank you for choosing LVC-2001. The fanless Box PC is an ideal vehicle gateway controller. LVC-2001 is built with Intel® Atom™ E3845/E3825 CPU, that supports 4 processor cores and operates with DDR3L SO-DIMM memory up to 4GB. The compact and fanless form factor make it suitable for vehicle applications. Regarding environmental reliability, LVC-2001 is certified with MIL-STD-810G vibration and shock resistance. As a vehicle computing system, LVC-2001 is designed with GPS and G-sensor I/O for navigation purpose. Other useful I/Os include 2 x RJ-45 LAN ports, 3 mini-PCIe expansion slots with 4 SIM card sockets, 2 serial ports, 1 optional CAN bus, 2 USB ports and +9~36VDC power input.

Features:

- Intel® Atom™ E3845 / E3825 CPU
- 1 x DDR3L SODIMM max up to 4 GB memory
- 2 x Intel® 10/100/1000 Mbps RJ-45 ports
- 2 x full-size mini-PCIe slots and 1 x half-size mini PCIe slot with total 4 external accessible SIM slots support Wi-Fi/3G/4G(LTE)/mSATA
- MIL-STD-810G vibration/shock resistance
- Display: 1 x VGA and 1 x HDMI
- USB: 1 x USB 3.0 type-A and 1 x USB 2.0 type-A
- Onboard GPS receiver module and G-sensor
- COM ports: 2 x RS-232/422/485 with RI/5V/12V
- Optional: CAN bus support J1939 / J1708

System Specifications

Dimensions		198x 52 x 185mm
Processor		Intel® BayTrail E3845/E3825
System Memory		1x DDR3L SO-DIMM, 4GB
BIOS		AMI SPI Flash BIOS
Storage	mSATA/SATA	1x mSATA Socket, 1x SATA 2.5" 9.5 mm drive bay
Ethernet Controller		2x Intel® i210-IT
Graphic Controller		Intel® HD Graphics
Audio		MIC IN/LINE OUT
I/O	LAN Port	2x RJ45 @ 10/100/1000Mbps
	Display Port	1x VGA Port 1x HDMI Port
	Serial I/O	COM1/2: 2x DB9 RS-232/422/485
	GPS	U-blox NEO-7N
	G-Sensor	ADXL 345
	USB	1x USB3.0 Type A 1x USB2.0 Type A
	Power Input	3-pin Terminal block (+, -, ignition), +9~36VDC, ATX mode support ignition delay On/Off control
	Expansion	2x Full-size mini-PCIe Socket (1x USB+PCIe+2x SIM; 1x USB+2x SIM) 1x Half-size mini-PCIe Socket
	CAN Bus	A1 SKU: 1x CAN Bus for J1939/J1708 on COM3 Port, A2 SKU: 1x CAN Bus (Optional for OEM)
	MIO	4x DI (3V or 12V TTL selectable), 4x DO (12V TTL, Max. 100mA) 2x MCU DI 1x 12V Output @ Max. 1A
Power	Input	3-pin Terminal Block (+, -, ignition), +9~36VDC
	Output	12V/1A DC Out
	Adaptor	Optional
Environment	Operating Temperature	-20~60°C (Industrial-grade components) -5~45°C (w/o Industrial-grade components)
	Humidity	5%~95%, non-condensing
Mechanical	System Design	Fanless
	Weight	1.8kg
	Mounting	Wallmount or Suspension Kit
Certification	EMC	CE/FCC Class A, E13, RoHS
Reliability	Compliance	Vibration/Shock: MIL-STD-810G, Method 514.6
OS Support	Windows	WES7 (WS7E)/W7 Pro SPI/ WE8 STD
	Linux	Kernel 2.6.18 or Later

Package Contents

Your package contains the following items:

- LVC-2001 Fanless Embedded System
- Power connector 3 pin x1 (P/N:04AW20031E001)
- HDD Screws x 4 (P/N: 070W102400602)
- Mini-PCle Screws x 8 (P/N: 070W101000401)
- MIO connector 16pin x1 (P/N: 04AW20161Z101)

Ordering Information

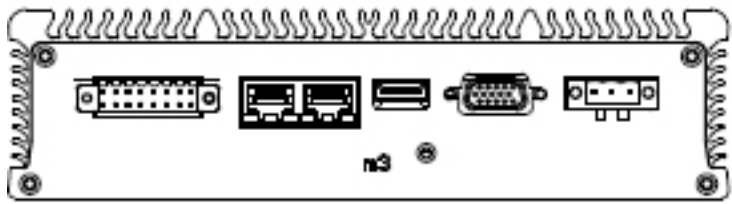
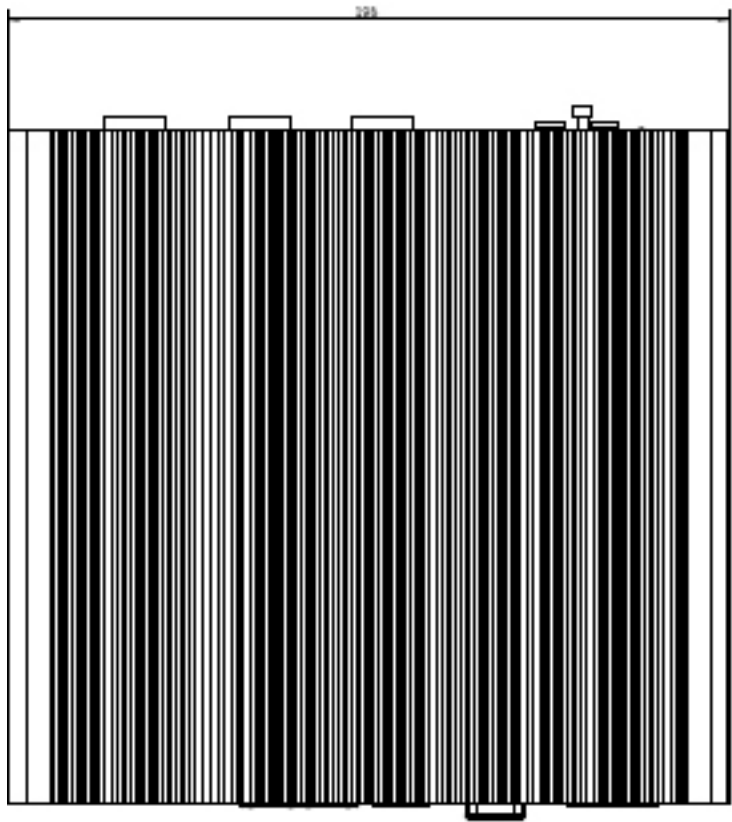
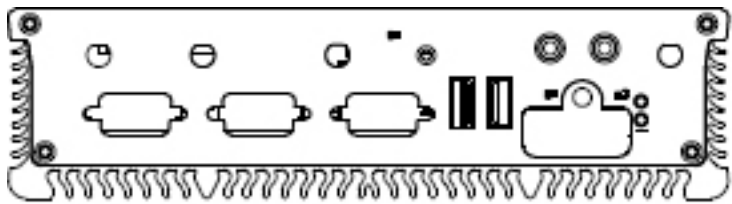
LVC-2001-A1	Intel® Atom SoC E3845 CPU In-Vehicle Computer, 1x DDR3L SODIMM, 3x Mini-PCle, 1x USB 3.0 host, 1x USB 2.0 host, 2x Serial port, 1x CAN Bus, DC Power input +9~36Vdc with Ignition
LVC-2001-A2	Intel® Atom SoC E3825 CPU In-Vehicle Computer, 1x DDR3L SODIMM, 3x Mini-PCle, 1x USB 3.0 host, 1x USB 2.0 host, 2x Serial port, 1x Optional CAN Bus, DC Power input +9~36Vdc with Ignition

Chapter 2:

System Components

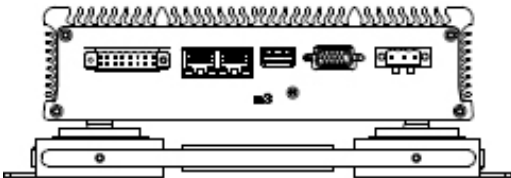
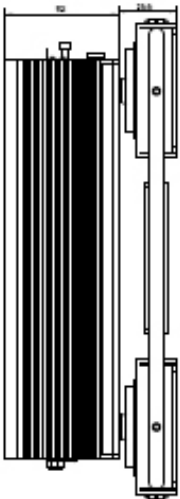
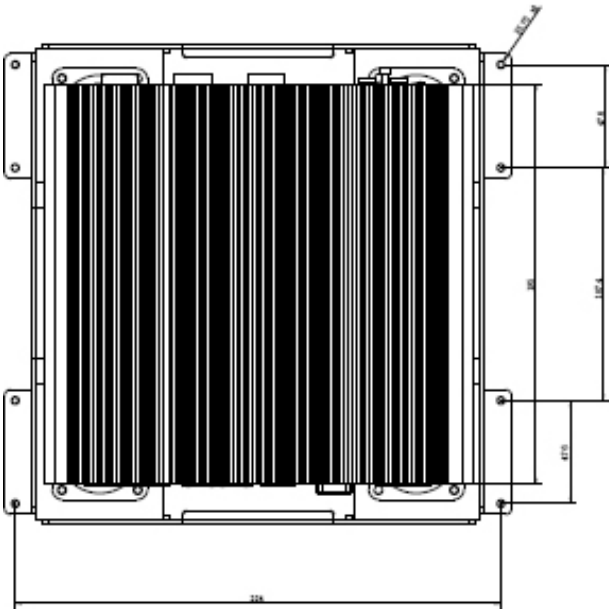
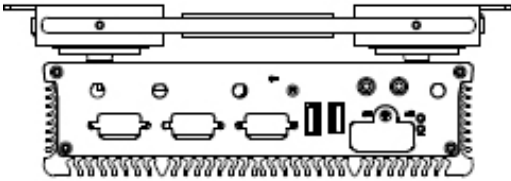
Mechanical Drawings

Mechanical Dimensions of the LVC-2001 with the System
Unit: mm



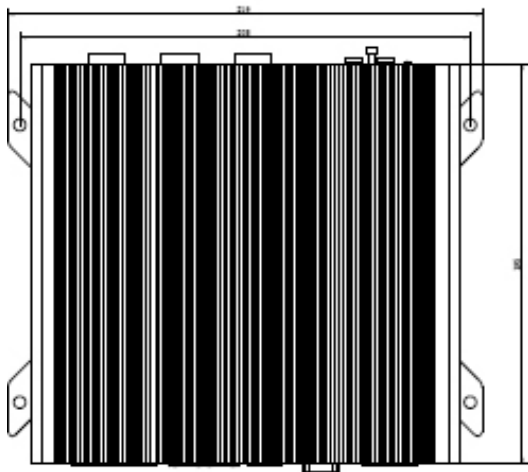
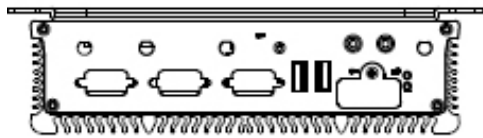
Mechanical Dimensions of the LVC-2001 with Vibration Kit

Unit: mm



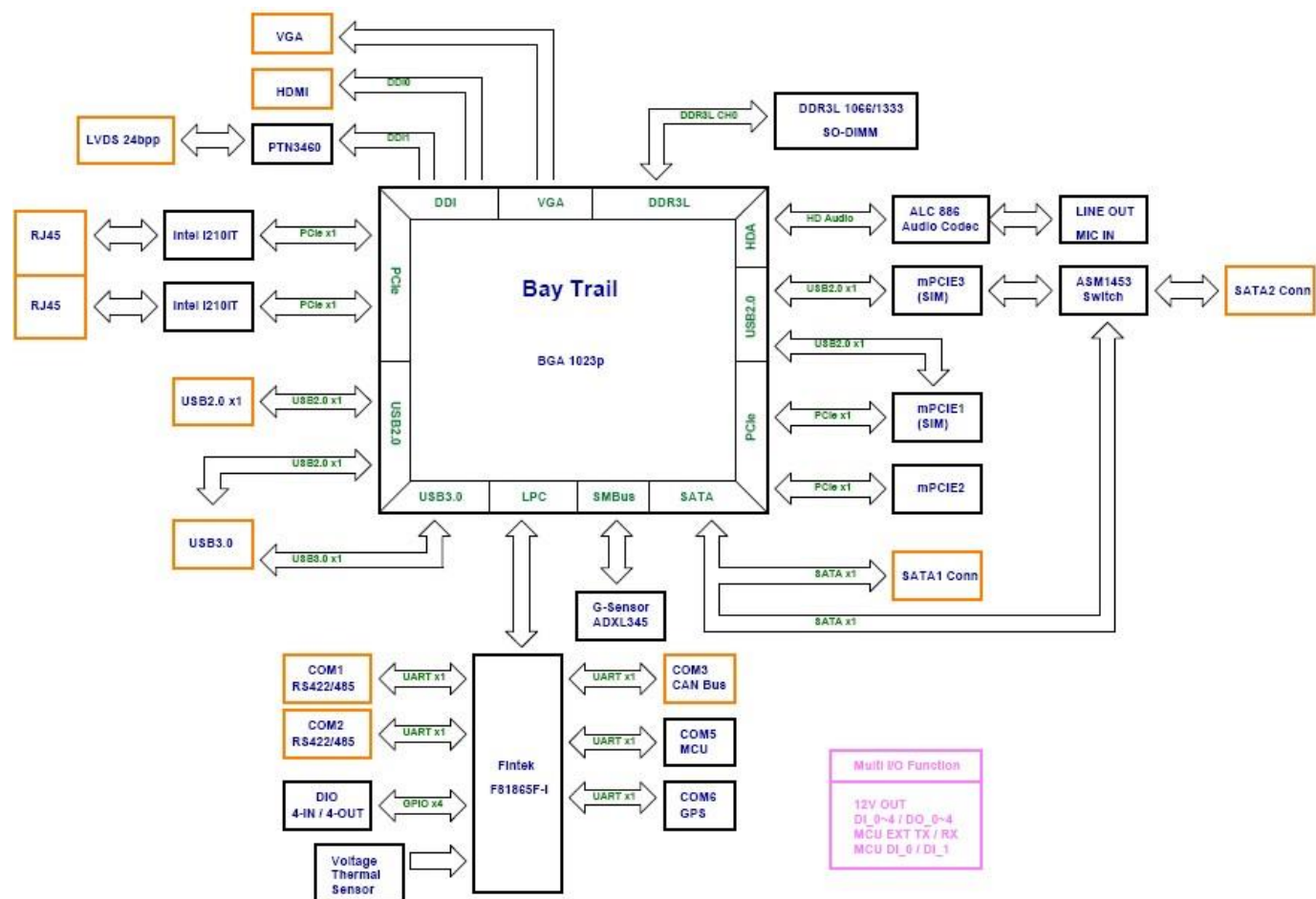
Mechanical Dimensions of the LVC-2001 with Wallmount Kit

Unit: mm



Block Diagram: The MainBoard

The block diagram depicts the relationships among the interfaces and modules on the motherboard.

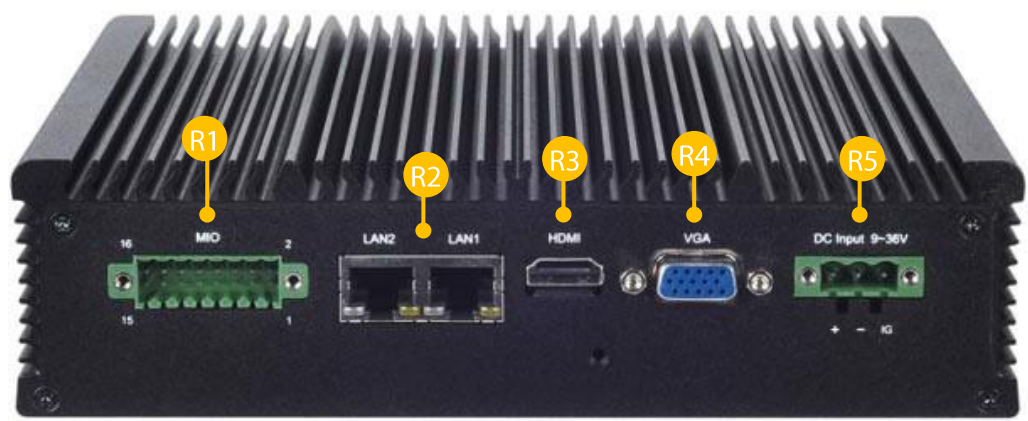


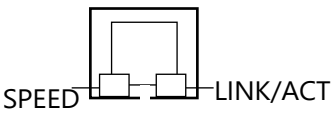
Front Panel



Component	Description
F1 LED	Power and HDD/SSD LED
F2 SIM	SIM card socket access with cover and lock-screw
F3 USB	1x USB 3.0 type A connector 1 x USB 2.0 type A connector
F4 CAN	An optional CAN bus supports J1939 &J1708 standards
F5 COM1/COM2	RS-232/422/485 ports for serial communication
F6 Audio	Mic IN/Line OUT

Rear Panel



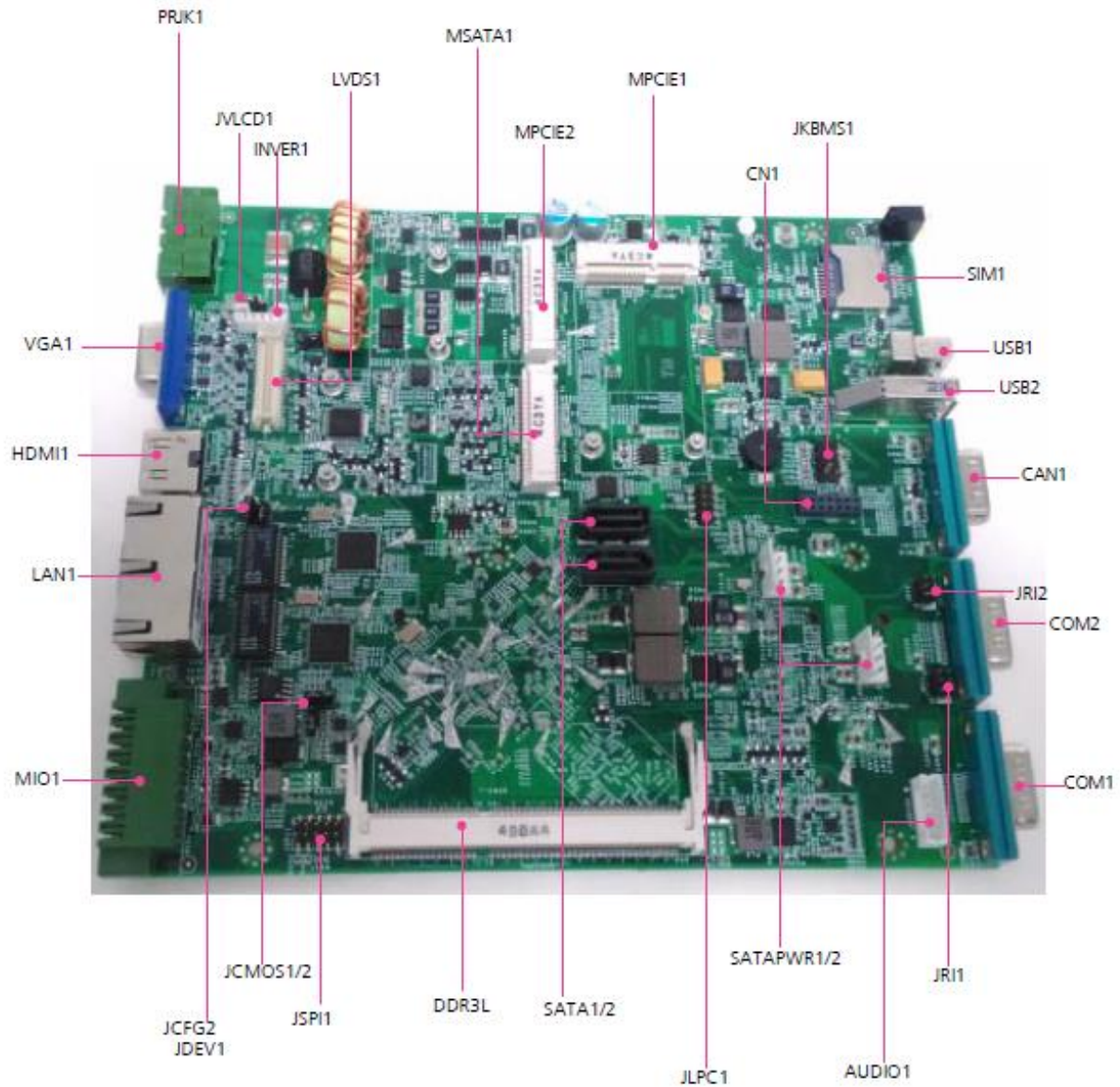
Component	Description
R1 Multiple-I/O Connector	A 16-pin male connector for the following functions: 4x DI (5V or 12V TTL selectable) 4x DO (12V TTL , Max. 100mA) 2x MCU DI 1x 12V Output @Max. 1A
R2 Two 10/100/1000Mbps LAN Ports 	Two RJ-45 (provided by Intel® i210IT) jacks with LED indicators as described below: LINK/ACT (Yellow) • On/Flashing: The port is linking and active in data transmission. • Off: The port is not linking. SPEED (Green/Amber) • Amber: The connection speed is 1000Mbps. • Green: The connection speed is 100Mbps • Off: The connection speed is 10Mbps.
R3 HDMI Port	1 x HDMI display port
R4 VGA Port	1 x VGA display port
R5 Power-Input (DC)	3-pin terminal block (+, -, ignition), +9~36VDC, ATX mode support ignition delay on/ off control

Chapter 3:

Motherboard Information

Connectors and Jumpers

The following picture highlights the locations of connectors and jumpers on the motherboard.

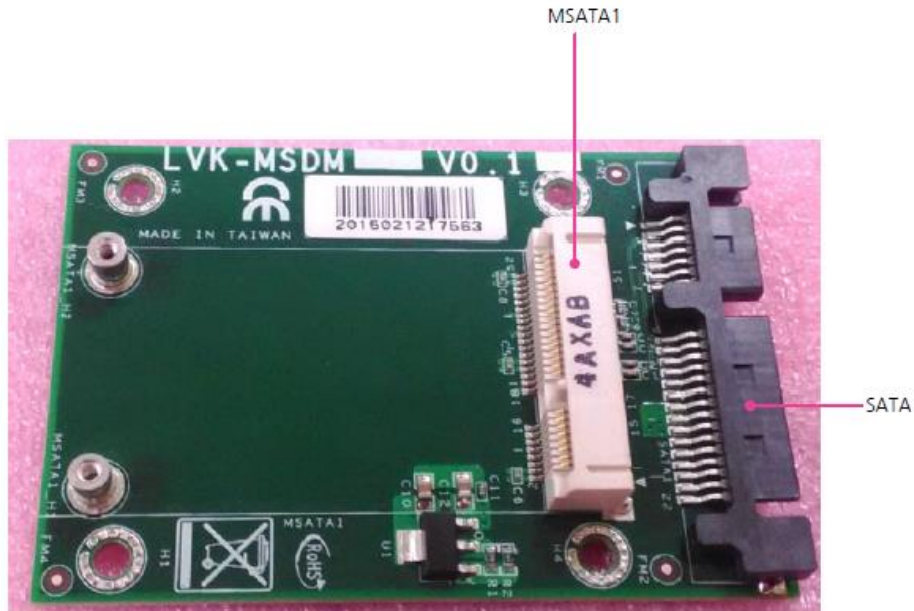




SIM2

Add-on Card: LVK-MSDM

LVC-2001 comes with a SATA-interface add-on card LVK-MSDM for extra mSATA storage (if the mSATA slot on the motherboard is occupied and assigned to function as a Mini-PCIe socket for other purposes).



Connectors and Jumpers List

The tables below list the function of each of the board jumpers and connectors by labels shown in the above section. The next section in this chapter gives pin definitions and instructions on setting jumpers.

Labels	Function
JCMOS1/2	Clear CMOS jumpers
JVLCD1	Jumper for LVDS setting
JMCU1	MCU programming jumper
JDEV1	EEPROM setting jumper
JCFG2	CFG jumper setting
VGA1	15-pin VGA port
HDMI1	HDMI display port
USB1	USB 2.0 Type-A port
USB2	USB 3.0 Type-A port
LAN1	2 x RJ-45 LAN ports
SATA1/2	2 x SATA 7-pin signal connectors
SATAPWR1/2	2 x SATA 4-pin power connectors
MSATA1	mSATA socket with USB signals, link to SIM1
SIM1/2	2 x SIM dual slots
JR1/2	COM1/COM2 power selection
MPCIE1	Full-sized mini-PCIe socket link to SIM2
MPCIE2	Half-sized mini-PCIe socket
COM1/2	2 x Serial COM ports
MIO1	16-pin multiple I/O connector
JBMS1	PS/2 keyboard and mouse connector
JLPC1	Low-pin count
JSPI1	SPI pin header
PRJK1	Power input
CN1	CAN bus female pin header
Inver1	LVDS inverter
CAN1	CAN bus connector using COM3 port
AUDIO1	Audio pin header

Jumper Settings

JCMOS1/2 (Clean CMOS Jumper):

This jumper is used to erase data in CMOS. To clear CMOS, first turn off your system and unplug power source. Then, by placing the cap on pin 2 and 3 (short pin 2-3), this jumper can erase the system settings stored in CMOS memory.



Pin	Description
Short 1-2	Normal (default)
Short 2-3	Clear RTC

JVLCD1: jumper for LVDS power setting



Pin	Description
Short 1-2	V3P3S
Short 2-3	V5S

JMCU1: Programming the MCU



Pin	Description
Short 1-2	Program MCU
Short 2-3	Normal (default)

JDEV1: EEPROM setting



Pin	Description
Short 1-2	EEPROM
Short 2-3	0x40h (default)

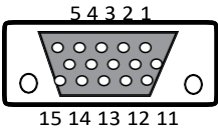
JCFG2



Pin	Description
Short 1-2	18bpp (default)
Short 2-3	24bpp

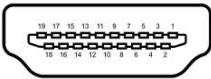
Connectors

VGA (VGA1)



Pin	Signal	Pin	Signal	Pin	Signal
1	RED	6	GND	11	N/A
2	GREEN	7	GND	12	DDC DAT
3	BLUE	8	GND	13	VGA_HS
4	N/A	9	V5S_VGA	14	VGA_VS
5	GND	10	GND	15	VGA_CLK

HDMI (HDMI1)



Pin	Description	Pin	Description
1	HDMI_DATAP2	2	GND
3	HDMI_DATAN2	4	HDMI_DATAP1
5	GND	6	HDMI_DATAN1
7	HDMI_DATAP0	8	GND
9	HDMI_DATAN0	10	HDMI_CLKP
11	GND	12	HDMI_CLKN
13	N/A	14	N/A
15	HDMI_DDC_CLK	16	HDMI_DDC_DAT
17	GND	18	V5S_HDMI
19	HDMI_HPD		

USB2.0 (USB1)



Pin	Description	Pin	Description
1	V5S_USB1	3	USBDP1
2	USBDN1	4	GND

USB3.0 (USB2)



Pin	Description	Pin	Description
1	V5S_USB2	5	USB3_SSRXN
2	USBDN2	6	USB3_SSRXP
3	USBDP2	7	GND
4	GND	8	USB3_SSTXP
		9	USB3_SSTXN

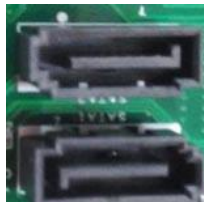
LAN (LAN1)



Pin	Description	Pin	Description
1	LAN1_MDI0P	13	LAN2_MDI0P
2	LAN1_MDI0N	14	LAN2_MDI0N
3	LAN1_MDI1P	15	LAN2_MDI1P
4	LAN1_MDI2P	16	LAN2_MDI2P
5	LAN1_MDI2N	17	LAN2_MDI2N
6	LAN1_MDI1N	18	LAN2_MDI1N
7	LAN1_MDI3P	19	LAN2_MDI3P
8	LAN1_MDI3N	20	LAN2_MDI3N
9	V3P3A	21	V3P3A
10	ACT_LAN1	22	ACT_LAN2
11	LNK1000_LAN1	23	LNK1000_LAN2
12	LNK100_LAN1	24	LNK100_LAN2

Serial-ATA Connectors (SATA1/2):

SATA 7-pin signal connector for HDD/SSD. The interface signal is SATA 3.0 Gbps.



SATA1		SATA2	
Pin	Description	Pin	Description
1	GND	1	GND
2	SATATXP	2	SATA2TXP
3	SATATXN	3	SATA2TXN
4	GND	4	GND
5	SATARXN	5	SATA2RXN
6	SATARXP	6	SATA2RXP
7	GND	7	GND

4-pin Serial-ATA Power Connector (SATAPWR1/2): It is for connecting the SATA power cord.

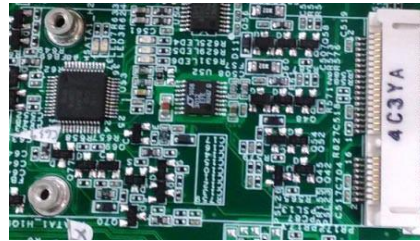
SATAPWR1/2



Pin	Description
1	VCC12
2	GND
3	GND
4	VCC5_PS

mSATA (MSATA1)

mSATA storage socket with USB and SIM signals, which may also allow this socket to function as a mini-PCIe connector for other purposes.



Pin	Description	Pin	Description
1	N/A	30	SMB_CLK
2	V3P3S	31	mSATATXN
3	N/A	32	SMB_DAT
4	GND	33	mSATATXP
5	N/A	34	GND
6	V1P5S	35	GND
7	N/A	36	USB_DN0
8	UIM1_PWR	37	GND
9	GND	38	USB_DP0
10	UIM_1_DAT	39	V3P3S
11	UIM_1_CLK	40	GND
12	UIM_1_RST	41	V3P3S
13	UIM_1_VPP	42	N/A
14	N/A	43	GND
15	GND	44	N/A
16	N/A	45	N/A
17	N/A	46	N/A
18	GND	47	N/A
19	N/A	48	V1P5S
20	N/A	49	N/A
21	GND	50	GND
22	PLTRST_BUF2_N	51	N/A
23	mSATARXP	52	V3P3S
24	V3P3S	53	N/A
25	mSATARXN	54	N/A
26	GND	55	N/A
27	GND	56	N/A
28	V1P5S	57	N/A
29	GND	58	N/A

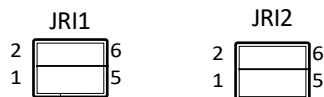
SIM card readers in Dual Slots (SIM1/2)



SIM1 Dual Slot		SIM2 Dual Slot	
Pin	Description	Pin	Description
1	UIM_PWR	1	UIM1_PWR
2	UIM1_RST	2	UIM3_RST
3	UIM1_CLK	3	UIM3_CLK
4	N/A	4	N/A
5	GND	5	GND
6	UIM1_VPP	6	UIM3_VPP
7	UIM1_DAT	7	UIM3_DAT
8	GND	8	GND
9	UIM_PWR	9	UIM1_PWR
10	UIM2_RST	10	UIM4_RST
11	UIM2_CLK	11	UIM4_CLK
12	N/A	12	N/A
13	GND	13	GND
14	UIM2_VPP	14	UIM4_VPP
15	UIM2_DAT	15	UIM4_DAT
16	GND	16	GND

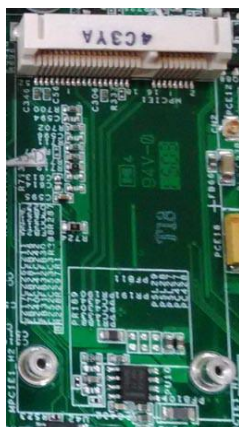
COM1/COM2 Power Selection (JRI1/JRI2):

JRI1 selects COM1 power voltage and JRI2 selects COM2 power voltage . The default is Ring Indicator (RI) for pin 8 of COM.



Pin No.	Signal
1-2	Default
3-4	VCC5
5-6	VCC12

MPCIE1: Mini-PCIe Connector



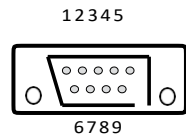
Pin	Signal	Pin	Signal
1	PCIE_WAKE_N	2	VCC3P3_PS
3	N/A	4	GND
5	N/A	6	V1P5_MPCIE
7	E_CLKREQ-	8	UIM_PWR
9	GND	10	UIM_DATA
11	PCIE_CKN3	12	UIM_CLK
13	PCIE_CKP3	14	UIM_RESET
15	GND	16	UIM_VPP
17	RSV	18	GND
19	RSV	20	N/A
21	GND	22	BUF_PLT_RST#
23	PCH_PCIE_RXN3	24	PCIE_PCIE_VCC3AUX
25	PCH_PCIE_RXP3	26	GND
27	GND	28	V1P5_MPCIE
29	GND	30	SMBCLK_RESUME
31	PCH_PCIE_TXN3	32	SMBDATA_RESUME
33	PCH_PCIE_TXP3	34	GND
35	GND	36	PCH_USB_N8
37	GND	38	PCH_USB_P8
39	VCC3P3_PS	40	GND
41	VCC3P3_PS	42	LED_WWAN1-
43	GND	44	LED_WLAN1-
45	RSV	46	N/A
47	RSV	48	V1P5_MPCIE
49	RSV	50	GND
51	RSV	52	VCC3P3_PS

MPCIE2: Mini-PCIe Connector (half-size)



Pin	Signal	Pin	Signal
1	WAKE#_MPCIE2	2	VCC3P3
3	N/A	4	GND
5	N/A	6	V1P5_MPCIE
7	CLKREQ3#_MPCIE	8	N/A
9	GND	10	N/A
11	PCIE_CLKN3_MPCIE	12	N/A
13	PCIE_CLKP3_MPCIE	14	N/A
15	GND	16	N/A
17	N/A	18	GND
19	N/A	20	N/A
21	GND	22	PLTRST_BUF3_N
23	PCIE_RXN3	24	V3P3S
25	PCIE_RXP3	26	GND
27	GND	28	V1P5
29	GND	30	SMBCLK_RESUME
31	PCIE_TXN3_MPCIE	32	SMBDATA_RESUME
33	PCIE_TXP3_MPCIE	34	GND
35	GND	36	N/A
37	GND	38	N/A
39	VCC3P3	40	GND
41	VCC3P3	42	N/A
43	GND	44	N/A
45	N/A	46	N/A
47	N/A	48	V1P5
49	N/A	50	GND
51	N/A	52	VCC3P3

COM1&2



Pin	Description
1	SP1_DCD
2	SP1_DSR
3	SP1_RXD
4	SP1_RTS
5	SP1_TXD
6	SP1_CTS
7	SP1_DTR
8	VCC_RI1
9	GND

Pin	Description
1	SP2_DCD
2	SP2_DSR
3	SP2_RXD
4	SP2_RTS
5	SP2_TXD
6	SP2_CTS
7	SP2_DTR
8	VCC_RI2
9	GND

Multiple I/O Connectors (MIO1): Multiple I/O pins for functions in serial communication, Digital In/Out, Ignition detection input for automatic wake-up function



Pin No.	Function
1	GND
2	12V_OUT
3	IGN_DI0
4	IGN_DI1
5	EXT_TXD_R
6	EXT_RXD_R
7	DI_0
8	DO_0
9	DI_1
10	DO_1
11	DI_2
12	DO_2
13	DI_3
14	DO_3
15	GND
16	GND

Keyboard & Mouse connector (JKBMS1)



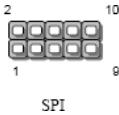
Pin No.	Pin Name	Pin No.	Pin Name
1	VCC5_KB	2	KCLK
3	MDATA		
5	KDATA		
7	GND	8	MCLK

Low-Pin Count (JLPC1)



Pin	Description	Pin	Description
1	33MHz CLK	2	LAD1
3	RESET	4	LAD0
5	FRAME	6	VCC 3.3
7	LAD3	8	GND
9	LAD2	10	GND

SPI (JSPI1)



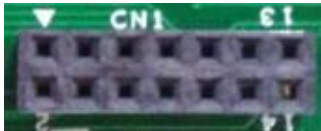
Pin	Description
1	SPI_HOLD
2	N/A
3	SPI_CS0
4	VCC3
5	SPI_MISO
6	N/A
7	N/A
8	SPI_CLK
9	GND
10	SPI_MOSI

Power Input with Ignition Control (PRJK1)



Pin No.	Pin Name
1	Ignition
2	GND
3	DC_IN

CAN Bus Module Connector (CN1)



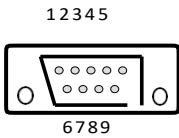
Pin No.	Signal	Pin No.	Signal
1	BAT_12V_24V	2	K_LINE
3	DO	4	N/A
5	GND_CAN	6	GND_CAN
7	PLTRST_BUF1	8	J1850+/J1708+
9	SIO_SIN3	10	J1850-/J1708-
11	SIO_SOUT3	12	CAN_H/J1939+
13	VSS	14	CAN_L/J1939-

LVDS Inverter (Inver1)



Pin No.	Pin Name
1	V12S
2	VSS
3	LVDS_BKLTCTL
4	LVDS_BKLTEN
5	GND

CAN Bus Connector (CAN1) on COM3 port



Pin No.	Pin Name
1	J1850-/J1708-
2	GND_CAN
3	CAN_H/J1939+
4	K_LINE
5	CAN_L/J1939-
6	J1850-/J1708-
7	J1850+/J1708+
8	J1850+/J1708+
9	BAT_12V_24V

AUDIOIN1: Line-out and Mic-in Connector



Pin No.	Pin Name
1	MIC_IN_L
2	MIC_IN_R
3	GND_AUO
4	GND_AUO
5	FRONT_OUT_L
6	FRONT_OUT_R

LVDS Connector (LVDS1)

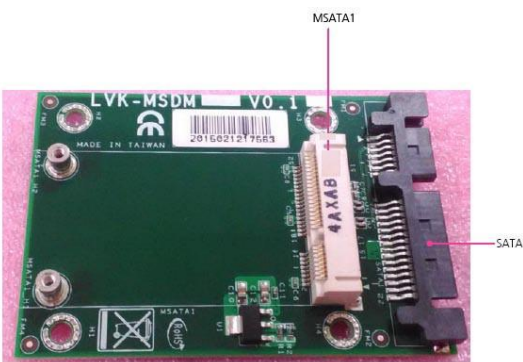
The 30-pin connector for LVDS display output



Pin	Description	Pin	Description
1	LVDS_VCC	2	LVDS_VCC
3	LVDS_VCC	4	LVDS_VCC
5	LVDS_VCC	6	LVDS_VCC
7	DDC_SCL	8	N/A
9	DDC_SDA	10	N/A
11	GND	12	N/A
13	LVDSA_DP0	14	LVDSA_DP3
15	LVDSA_DN0	16	LVDSA_DN3
17	GND	18	N/A
19	LVDSA_DP1	20	N/A
21	LVDSA_DN1	22	N/A
23	GND	24	GND
25	LVDSA_DP2	26	LVDSA_CKP
27	LVDSA_DN2	28	LVDSA_CKN
29	GND	30	GND

mSATA (MSATA1)

mSATA storage socket on LVK-MSDM add-on card.



Pin	Description	Pin	Description
1	N/A	30	N/A
2	V3P3S	31	SATA_TX_C_DN
3	N/A	32	N/A
4	GND	33	SATA_TX_C_DP
5	N/A	34	GND
6	N/A	35	GND
7	N/A	36	N/A
8	N/A	37	GND
9	GND	38	N/A
10	N/A	39	V3P3S
11	N/A	40	GND
12	N/A	41	V3P3S
13	N/A	42	N/A
14	N/A	43	GND
15	GND	44	N/A
16	N/A	45	N/A
17	N/A	46	N/A
18	GND	47	N/A
19	N/A	48	V1P5S
20	N/A	49	N/A
21	GND	50	GND
22	N/A	51	N/A
23	SATA_RX_C_DP	52	V3P3S
24	V3P3S	53	N/A
25	SATA_RX_C_DN	54	N/A
26	GND	55	N/A
27	GND	56	N/A
28	N/A	57	N/A
29	GND	58	N/A

SATA Male Connector (SATA1)

SATA male connector on LVK-MSDM add-on card.



Pin	Description	Pin	Description
1	GND	12	GND
2	SATA_TX_C_DP	13	GND
3	SATA_TX_C_DN	14	V5S
4	GND	15	V5S
5	SATA_RX_C_DN	16	V5S
6	SATA_RX_C_DP	17	GND
7	GND	18	N/A
8	N/A	19	GND
9	N/A	20	N/A
10	N/A	21	N/A
11	GND	22	N/A

Chapter 4:

Hardware Setup

Preparing the Hardware Installation

To access some components and perform certain setup, please read the warning below before installation procedures.

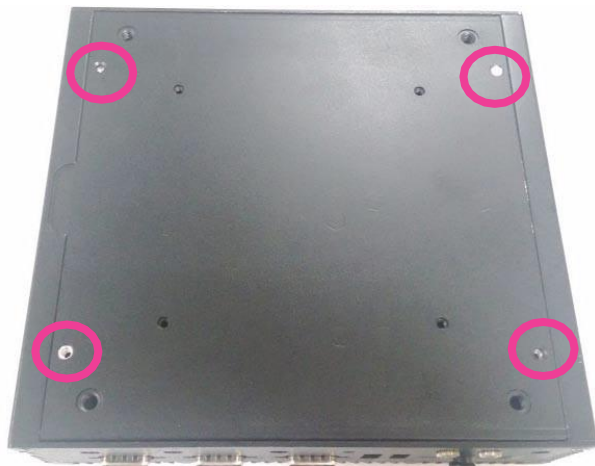


WARNING:

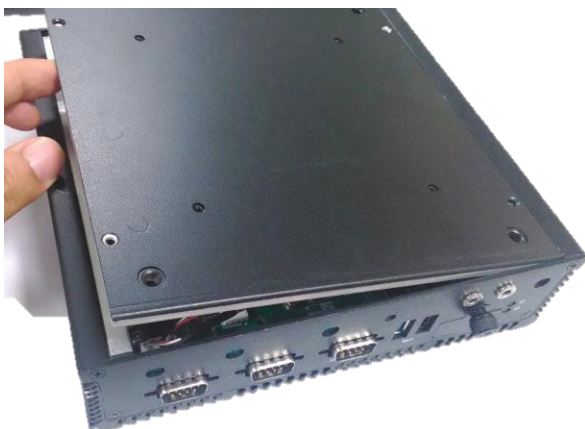
To reduce the risk of personal injury, electric shock, or damage to the equipment, please power off the system and remove all power connections.

Please wear ESD protection gloves before conducting the following steps.

1. Make sure LVC-2001 is powered off and disconnected from any power source.
2. Turn the device upside-down.
3. Remove the 4 screws circled in the image below.



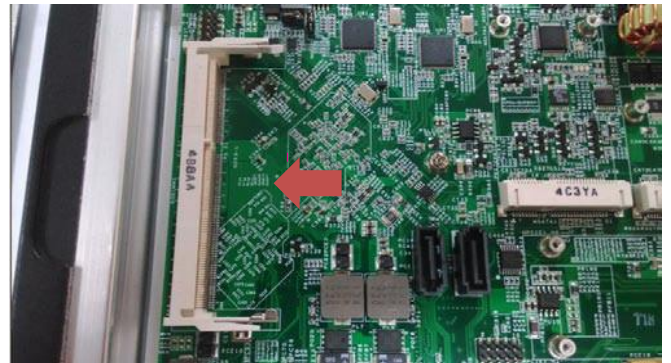
4. Apply force at the opening notch to lift the compartment.



Installing Memory Module

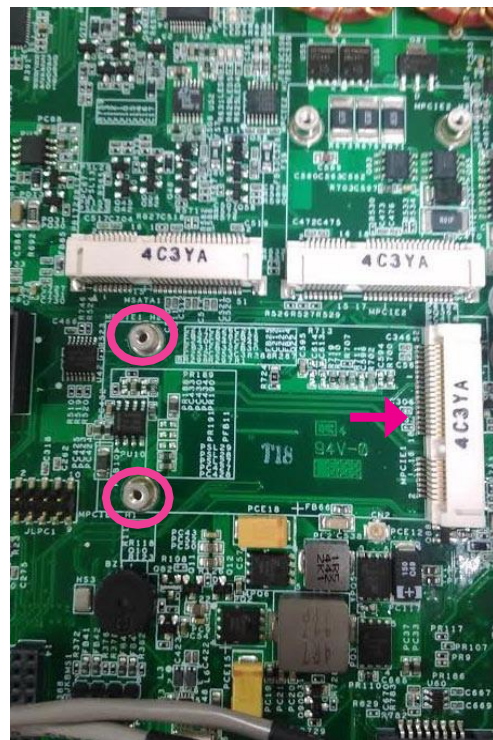
1. Locate the DDR socket on the motherboard.
2. Insert the DDR module to the socket. Make sure the notches are aligned.
3. Press the module down and the latches will clip on the module.

Note: the system supports DDR3L SO-DIMM x1 up to 4GB memory. Please use memories within the defined specifications to ensure proper operations.



Installing Full-sized Mini-PCIe Module

1. Align the Mini-PCIe module's keys with the Mini-PCIe slot (MPCIE1/MSATA1) notch.
2. Insert the module into the connector.
3. Press the module down and install the module with screws.



Installing mSATA Storage Device

1. Align the mSATA storage device with the MSATA1 socket.
2. Insert the device into the socket.
3. Press the device down and secure it with two screws.

Note: this MSATA1 socket also supports full-sized Mini-PCIe module for other applications.



Installing a SATA/mSATA Storage Device

The system provides one SATA 2.5" HDD/SSD drive bay and an additional mSATA socket (if the mSATA slot on the motherboard is occupied and assigned to function as a Mini-PCIe socket for other purposes). Both are attached on the back of the compartment.

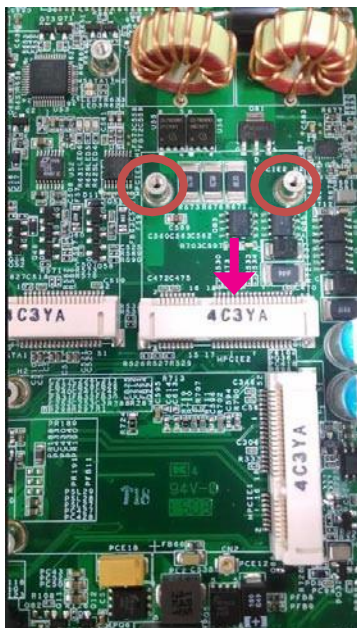
Please follow the steps below for installing a SATA/mSATA storage device.



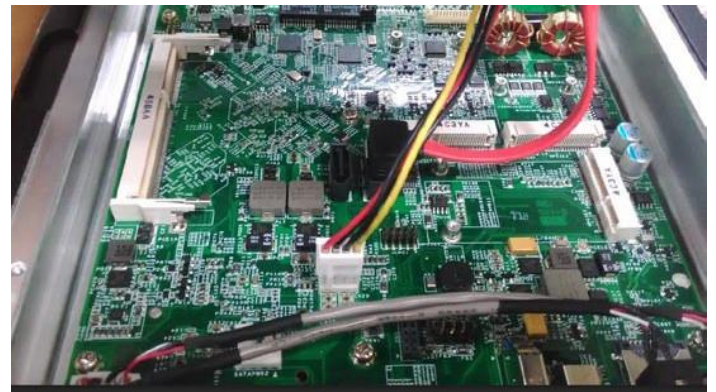
1. Locate the SATA signal and power connectors on the motherboard.

Installing Half-sized Mini-PCIe Module

1. Align the Mini-PCIe module's keys with the Mini-PCIe slot (MPCIE2) notch.
2. Insert the module into the connector.
3. Press the module down and install the module with screws.

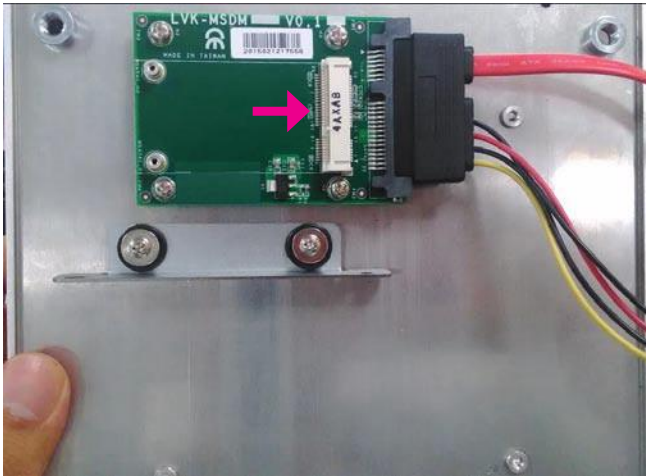


2. Insert a SATA cable into the SATA connectors on the board, as shown in the image below.



To Install a mSATA Storage Device

Connect the other end of the SATA cable to the SATA connector of LVK-MSDM add-on card. Then insert a mSATA storage device into the mSATA socket.



Connecting Power

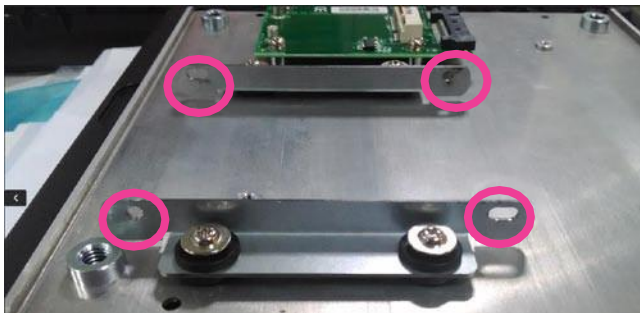
Connect the LVC-2001 to a +9V ~ +36V vehicle battery. The DC power-in connector comes with a 3-pin terminal block for its Phoenix contact. This power socket can only accept the power supply with the right pin contact so be cautious when inserting power to the system.



To Install a SATA 2.5" HDD/SSD

Place a SATA 2.5" HDD/SSD onto the drive bay.

Use 4 screws to secure the drive.



Connect the other end of the SATA cable

Note: The bottom cover needs to be re-assembled in the same direction as it was taken out to avoid HDD and cables interference.

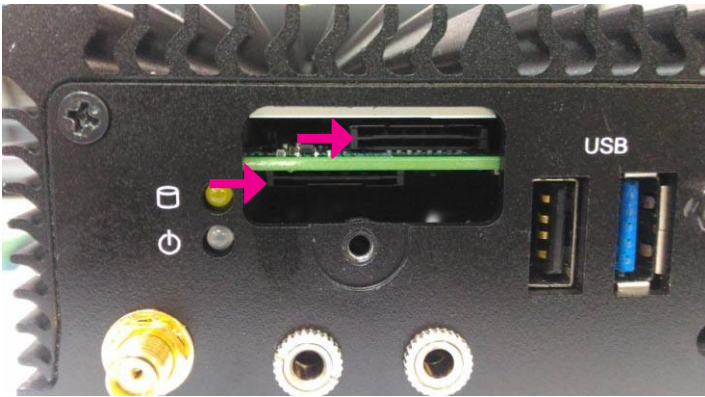
Installing SIM Cards

The system is designed with externally accesible SIM card slots. Please follow the steps below for installations.

1. Locate the door of SIM card slot and loosen the captive screw.



2. You will see 4 SIM slots; two at the upper side and the other two at the bottom side.



3. Insert SIM card.

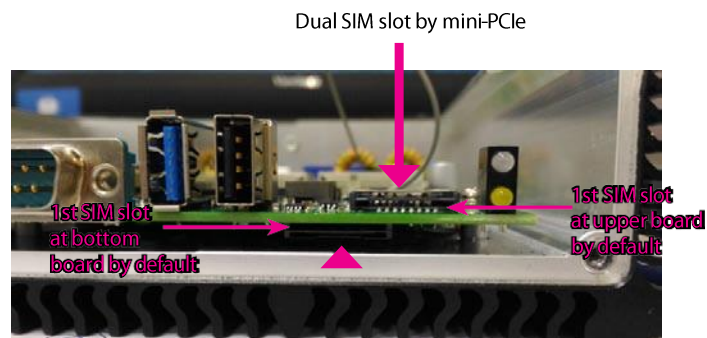


4. Place the door back and tighten the captive screw.



Notes:

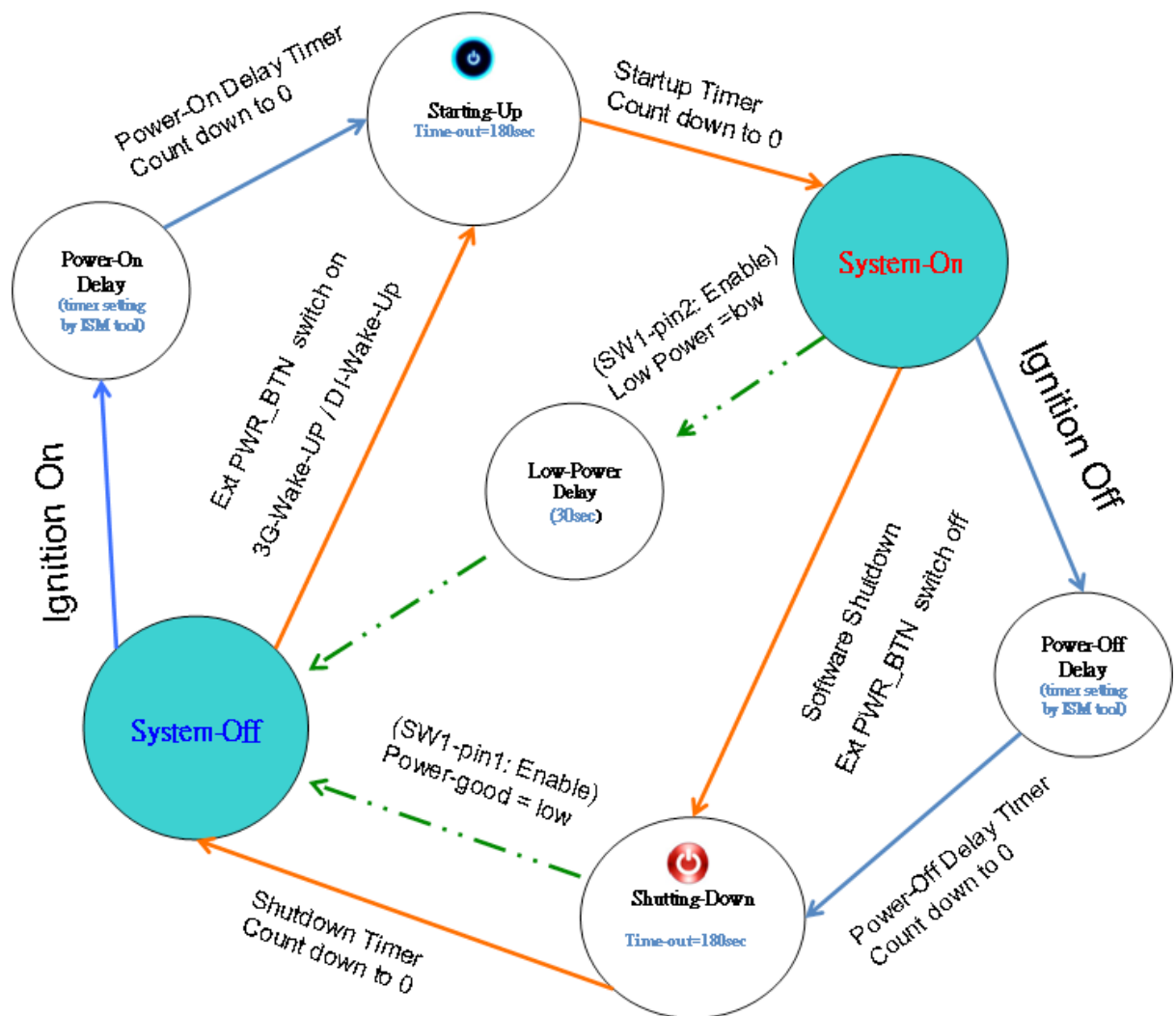
There are two dual-SIM slot sets. The dual SIM slots at the upper side of the motherboard are based on mini-PCIe interface while the bottom dual SIM slots are based on mSATA interface. Please refer to the image below for details.



Dual SIM slot by mSATA

Chapter 5: The Flow Chart

The flow chart section contains all flow chart used in the system. The flow chart describes the system's behavior on powering on and off the system via power ignition control or on/off switch when the appropriate timer control parameters are set.



Note:

1. For power-good and low-voltage mechanism to function to the workflow, you will need to enable the power-good and low-voltage detection function with selector1 and selector2 jumper respectively of SW1. (Refer to Chapter 3 Board Layout).
2. For power on and power off delay timer parameter, refer to Appendix A Using the Ignition System Manager (ISM).
3. For DI wake-up function, refer to "MIO" in the chapter of "Motherboard Information." Refer to Chapter 3 Board Layout and Appendix A Using the Ignition System Manager (ISM) for jumper setting and parameter setting respectively.
4. When the system's shutdown timer start counting down 180sec, using ignition or External PWR_BTN to start the system again during shutdown process will not work until the countdown finishes.

Appendix A:

Using the Ignition System Manager (ISM)

The Ignition System Manager (ISM) is a software that can monitor the system's voltage level and configure the features that the Power Ignition Module provides.

For sample ISM code, see *ISM* folder under LVC-2001 Utility on the *Driver and Manual CD*.

Running the Program

Just double click the ISM.exe to launch the ISM.

The program can configure the following values:

Voltage: It shows the current power system.

Power Input System: Select either 12V or 24V for vehicular power input.

Startup Voltage (V): If the DC-in voltage is not higher than this value, the system will not be able to start up.

Shutdown Voltage (V): If the DC-in voltage is lower than the shutdown voltage, the system will start shutdown process automatically. (Refer to selector 2 of SW1 dip switch on the mainboard.)

Power-on Delay (min/sec): Select power-on delay value to indicate the time to delay powering on the system. (Refer to the flow chart in Chapter 4)

Power-off Delay (hr/min/sec): Select power-off delay value to indicate the time to delay powering off the system (Refer to the flow chart in Chapter 4)

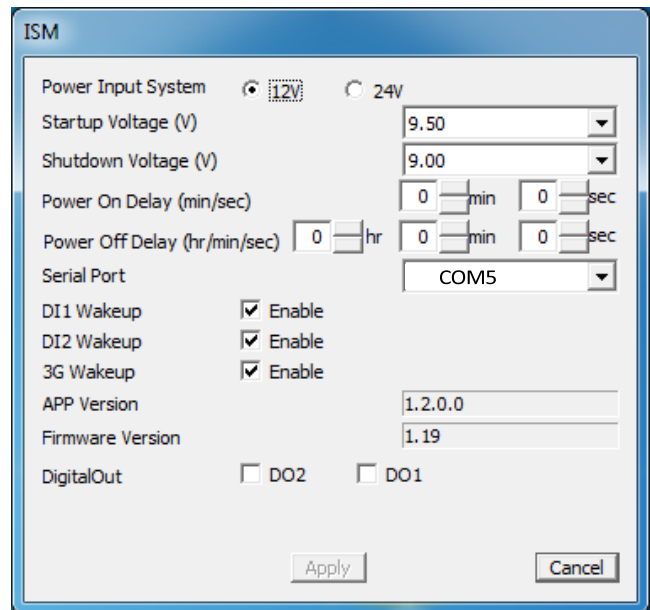
Serial Port: Select the serial communication port for the ISM. Choose COM5.

D1/D2 Wakeup: Digital input triggering to enable automatic wake-up function. Select this option and it will start the system automatically once an input has been triggered.

3G Wakeup: 3G SMS/Ring wake-up to enable automatic wake-up function. Select this option and it will start the system automatically through 3G Internet service.

DigitalOut: Check the box to turn on the output device and check off the box to turn off the connected device.

After you have made changes, click **Apply** to apply the changes to the Ignition controller or **Cancel** to cancel the changes.



Note:

1. You will have to enable (the default is enabled) the *selector 2 (Low Voltage Detection) of SW1 dip switch* on the mainboard to enable automatic shutdown function. (Refer to *Select MCUDetect Function for power ignition behavior (SW1) in Motherboard Information.*)
2. DI1/DI2 Wakeup function is detected via MIO pins (pin 9/11) (Refer to *MIO in Motherboard Information.*)
3. DO1 function is connected (controlled) via pin 10 while DO2 is connected (controlled) via pin 12. (Refer to *MIO in Motherboard Information.*)
4. Refer to the flow charts in Chapter 4 for more information.

Appendix B:

Digital Input/Output

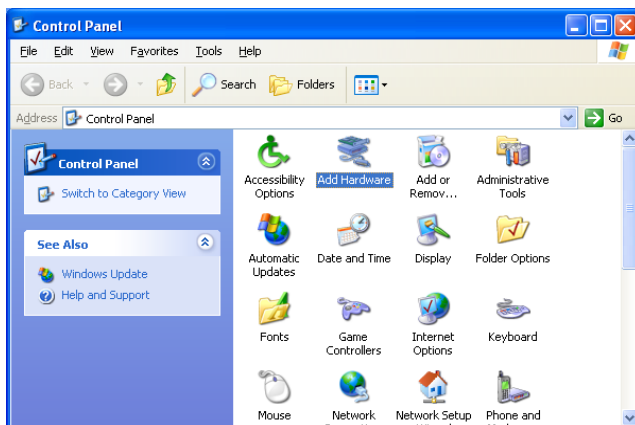
The Digital I/O on the rear panel is designed to provide the input and output operations for the system. For sample DIO code, see SuperIO folder under LVC-2001 Utility on the *Driver and Manual CD*. Make sure that you have installed the Lanner GPIO driver as instructed below.

Driver Installation

Before you could access or control the operation of the G-sensor, GPS and Digital I/O functions, install the the L_IO driver which is the library and driver needed for Lanner General Purpose Input/Output interface or functions.

To install the L_IO driver:

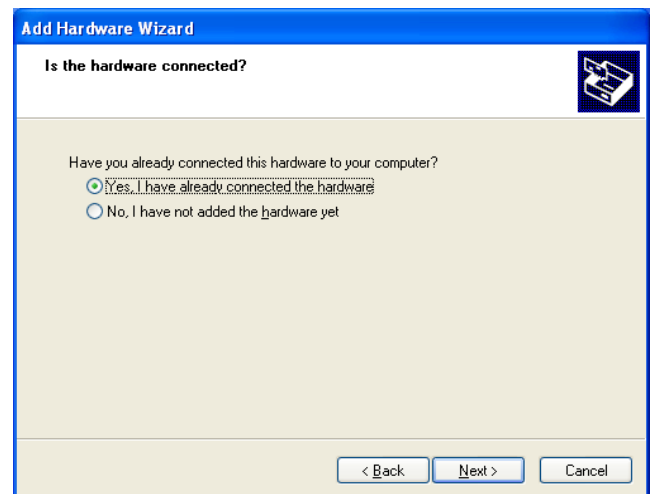
1. Restart the computer, and then log on with Administrator privileges.
2. Insert the Drivers and User's Manual CD to the USB-optical drive.
3. Browse the contents of the support CD to locate the file in the LIO folder.
4. From the control panel, click the ADD Hardware program



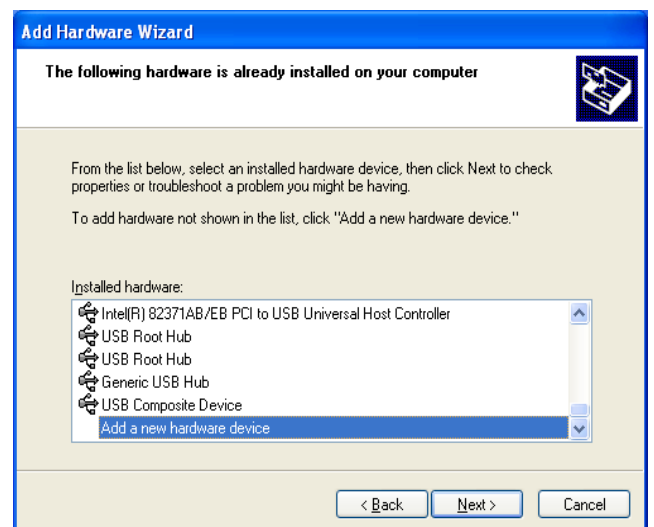
5. Select Next to proceed



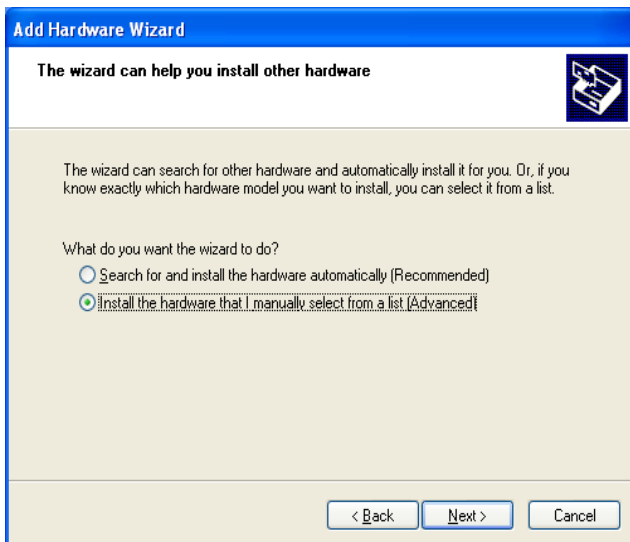
6. Answer "Yes" to the question and select Next to proceed.



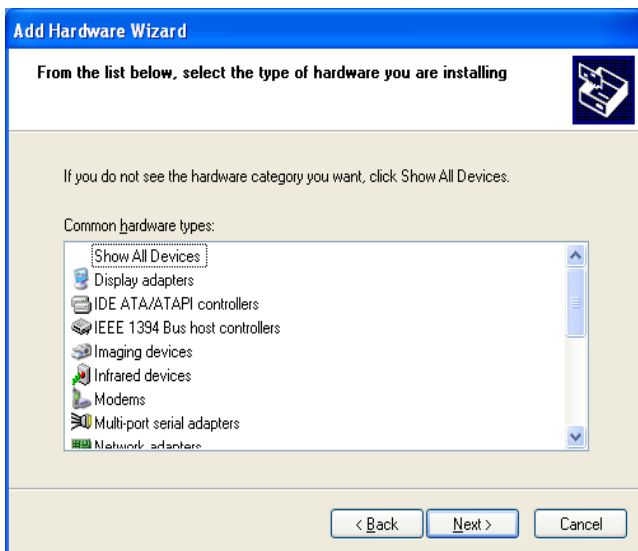
7. Select Add a new hardware device.



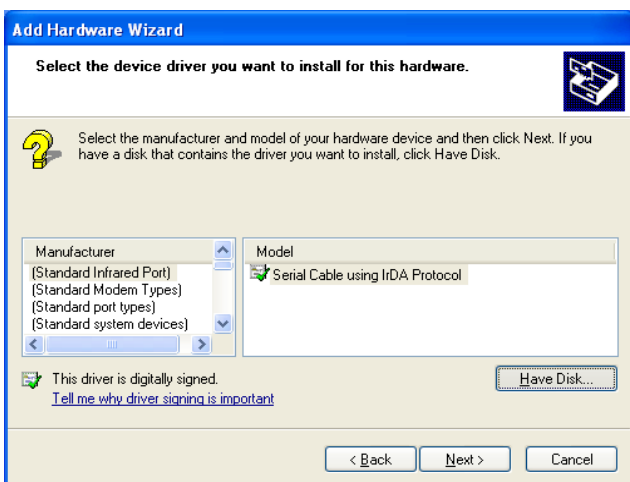
8. Choose to select the hardware Manually



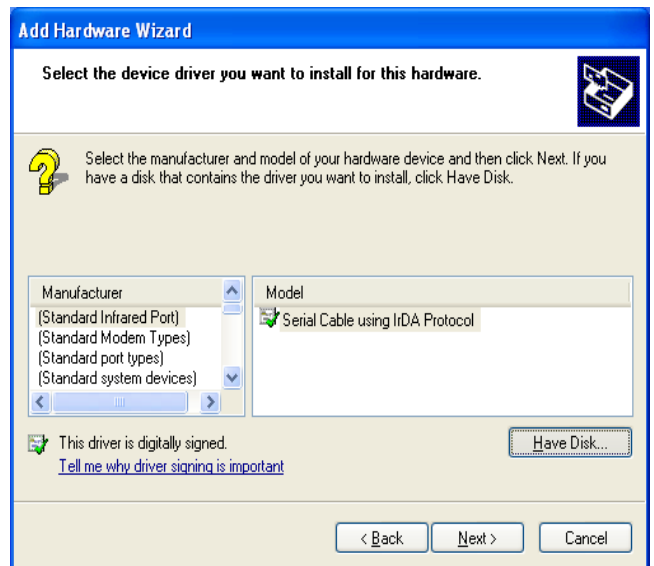
9. Choose Show all device and click Next.



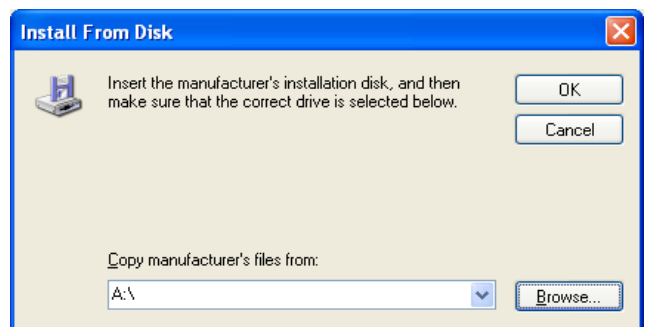
10. Click HaveDisk to locate the L_IO.inf file



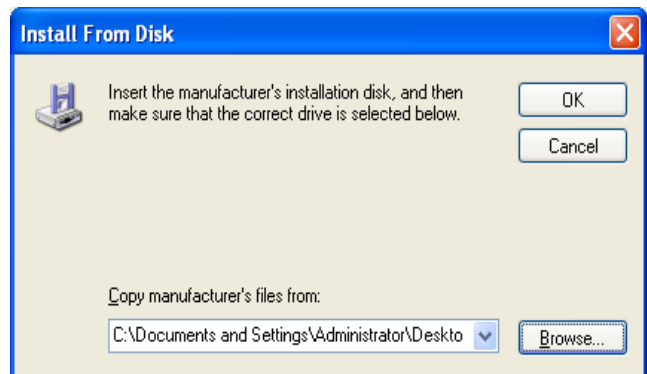
11. Click HaveDisk to locate the L_IO.inf file



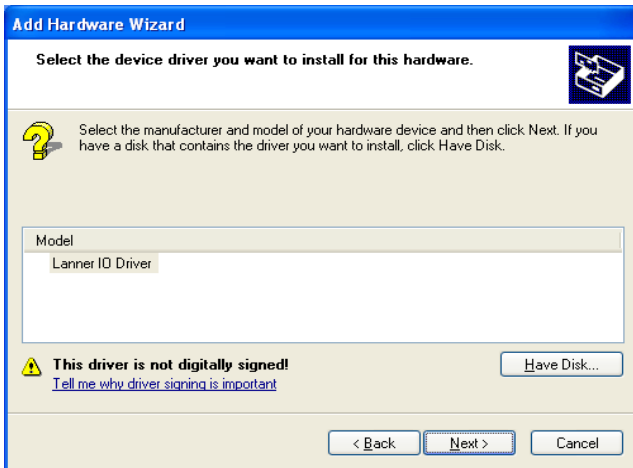
12. Select the L_IO.inf



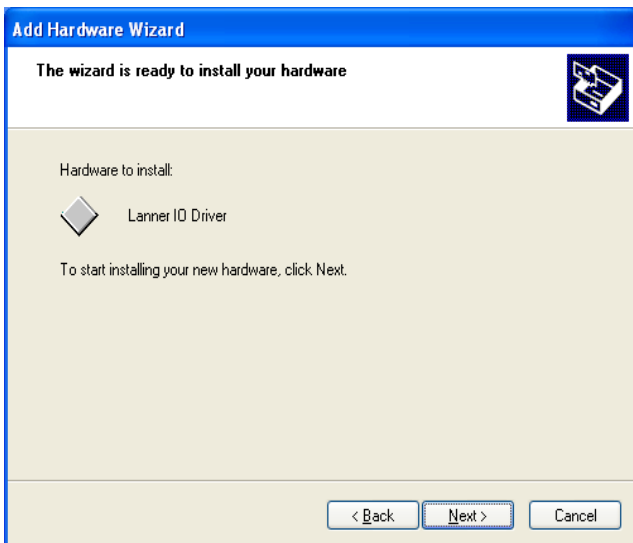
13. Select OK to confirm with the installation



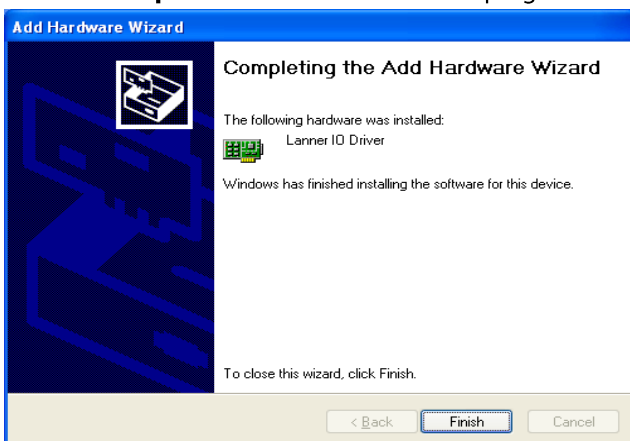
14. Select the Lanner IO driver and click Next.



15. Click Next



16. Click **Complete** to close the installation program.



To verify the GPIO driver installation, do the following steps:

1. Right-click on the My Computer icon, and then select Properties from the menu.
2. Click the Hardware tab, then click the Device Manager button.
3. Click the + sign next to the Lanner_Device, then the Lanner IO Driver should be listed.



A sample DIO program in C:

ioaccess.c: IO access code for Lanner Platform Digital IO program

```
*****  
*****/
```

```
#include "../include/config.h"
```

```
#ifndef DJGPP
```

```
/* standard include file */
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
/* For DOS DJGPP */
```

```
#include <dos.h>
```

```
#include <inlines/pc.h>
```

```
#else //DJGPP
```

```
/* For Linux */
```

```
#ifdef DIRECT_IO_ACCESS
```

```
/* For Linux direct io access code */
```

```
/* standard include file */
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

```
#if defined(LINUX_ENV)
```

```
#include <sys/io.h>
```

```
#endif
```

```
#if defined(FreeBSD_ENV)
```

```
#include <machine/cpufunc.h>
```

```
#endif
```

```
#include <time.h>
```

```
#include <stdint.h>
```

```
#include <fcntl.h>
```

```
#include <errno.h>
```

```
#include <string.h>
```

```
#define delay(x) usleep(x)
```

```
#endif
```

```
#ifndef MODULE
```

```
#include <linux/kernel.h>
```

```
#include <linux/module.h>
```

```
#include <linux/kernel.h>
```

```
#include <linux/fs.h>
```

```
#include <asm/io.h>
```

```
#include <linux/delay.h>
```

```
#undef delay
```

```
#define delay(x) mdelay(x)
```

```
#undef fprintf
```

```
#define fprintf(S, A) printk(A)
```

```
#endif //MODULE
```

```
#ifndef KLD_MODULE
```

```
#include <sys/types.h>
```

```
#include <sys/param.h>
```

```
#include <sys/systm.h>
```

```
#include <sys/malloc.h>
```

```
#include <sys/kernel.h>
```

```
#include <sys/bus.h>
```

```
#include <sys/errno.h>
```

```
#include <machine/bus.h>
#include <machine/resource.h>
```

```
#endif
```

```
#endif
```

```
/* local include file */
#include "../include/ioaccess.h"
```

```
#if (defined(MODULE) || defined(DIRECT_IO_ACCESS) ||
defined(KLD_MODULE))
```

```
/*
*-----
*
* LEB-5000 Version V1.0
*output3-0 = GPIO 03-00, input3-0= GPIO 53-50
*-----
*
*/
```

```
/*
* Device Depend Definition :
*/
```

```
#define INDEX_PORT    0x2E
#define DATA_PORT    0x2F
```

```
void enter_SIO_config(void)
{
    outportb(INDEX_PORT,0x87); // Must Do It Twice
    outportb(INDEX_PORT, 0x87);
    return;
}
```

```
void exit_SIO_config(void)
{
```

```
    outportb(INDEX_PORT,0xAA);
    return;
}

unsigned char read_SIO_reg(int LDN, int reg)
{
    outportb(INDEX_PORT,0x07); //LDN register
    delay(5);
    outportb(DATA_PORT, LDN);
    delay(5);
    outportb(INDEX_PORT, reg);
    delay(5);
    return(inportb(DATA_PORT));
}
```

```
void write_SIO_reg(int LDN, int reg, int value)
{
    outportb(INDEX_PORT,0x07); //LDN register
    delay(5);
    outportb(DATA_PORT, LDN);
    delay(5);
    outportb(INDEX_PORT, reg);
    delay(5);
    outportb(DATA_PORT,value);
    return;
}

void dio_gpio_init(void)
{
    enter_SIO_config();
    write_SIO_reg(0x6, 0x30,0x01);    //enable GPIO
Port
    write_SIO_reg(0x6,0xf0,((read_SIO_reg(0x6,
0xf0)& 0xf0)|0x0f)); //RxF0[3-0]=1111b, output
    write_SIO_reg(0x6,0xA0,(read_SIO_reg(0x6,
0xA0)& 0xf0)); //RxA0[3-0]=0000b,      input
```

```

        exit_SIO_config();
    return;
}

void dio_set_output(unsigned char out_value)
{
    enter_SIO_config();
    write_SIO_reg(0x6,0xf1,((read_SIO_reg(0x6,
0xf1)& 0xF0)|out_value));
    exit_SIO_config();
    return;
}

unsigned int dio_get_input(void)
{
    unsigned int tmp=0x00;
    enter_SIO_config();
    tmp=read_SIO_reg(0x6,0xA2)&0x0f;
    exit_SIO_config();
    return tmp;
}

//=====
=====
=====

#endif

```

Appendix C:

Accessing the GPS Data from the LVC-2001

The LVC-2001 employs an onboard u-blox NEO-7N GPS module for vehicle tracking and navigation system. You could read the GPS data through the RS-232 serial port.

It has the following listed key features and performance ratings:

Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS
Time-To-First-Fix (All satellites at -130 dBm)	Cold Start: 26 s Warm Start: 26 s Hot Start: 1 s Aided Starts: 1 s
Sensitivity	<ul style="list-style-type: none">• Tracking & Navigation: -162dBm• Reacquisition: -160dBm• Cold Start (without aiding): -148 dBm• Hot Start: -157 dBm
Maximum Navigation update rate	5Hz
Horizontal position accuracy (CEP, 50%, 24 hours static, -130dBm, SEP: <3.5m)	GPS: 2.5m SBAS: 2.0m
Configurable Timepulse frequency range	0.25 Hz to 1 kHz
Accuracy for Timepulse signal	RMS: 30 ns 99%: <60 ns Granularity: 21 ns Compensated: 15 ns
Velocity accuracy	0.1m/s
Heading accuracy	0.5 degrees

Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS
Time-To-First-Fix (All satellites at -130 dBm)	Cold Start: 26s Warm Start: 26s Hot Start: 1s Aided Starts: 1s
Operational Limits	Dynamics: less than and equal to 4g Altitude: 50,000m Velocity: 500m/s (Assuming Airborne <4g platform)

Specify the following communication parameters:

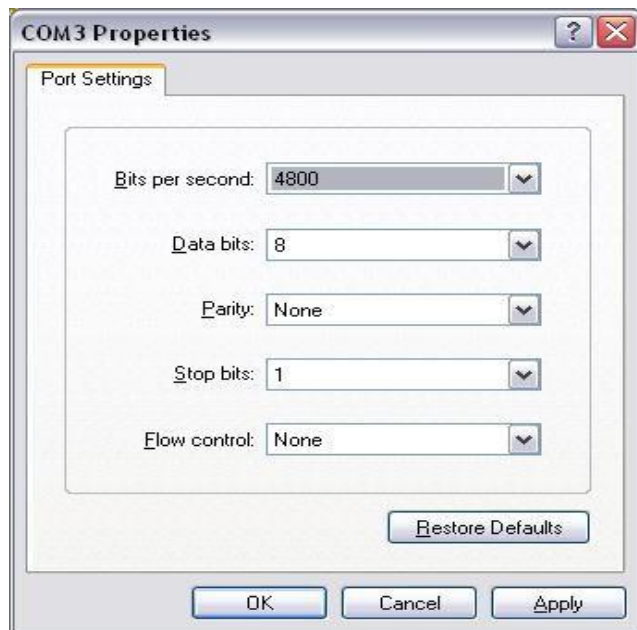
Bits per Second: 9600

Data Bits: 8

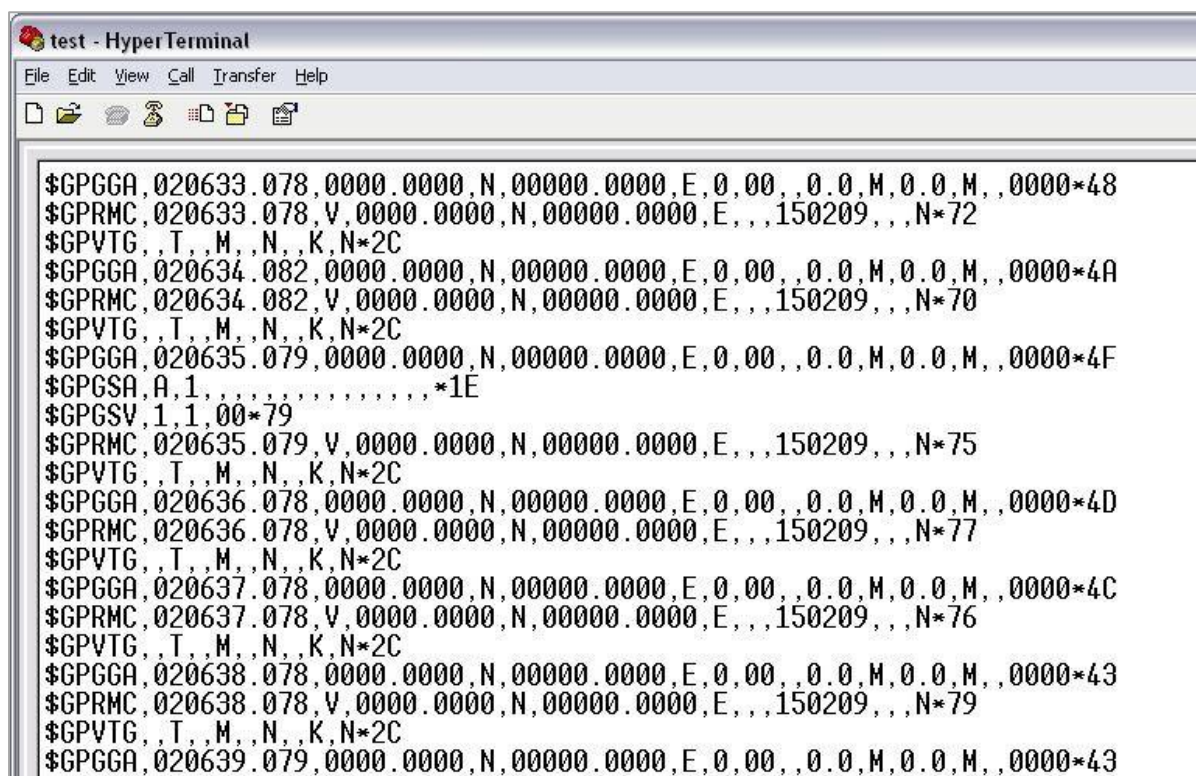
Parity: None

Stop Bit: 1

Flow Control: None



The hyper terminal should display GPS data:

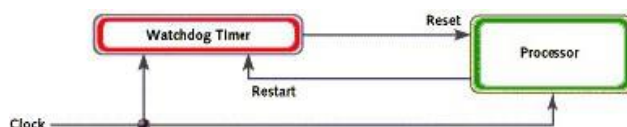


Appendix D:

Programming System Watchdog Timer of the LVC-2001

A watchdog timer is a piece of hardware that can be used to automatically detect system anomalies and reset the processor in case there are any problems. Generally speaking, a watchdog timer is based on a counter that counts down from an initial value to zero. The software selects the counter's initial value and periodically restarts it. Should the counter reach zero before the software restarts it, the software is presumed to be malfunctioning and the processor's reset signal is asserted. Thus, the processor will be restarted as if a human operator had cycled the power.

For sample watchdog code, see *watchdog* folder under LVC-2001 Utility on the *Driver and Manual CD*



Executing through the Command Line:

Execute the WD.EXE file under DOS (WD.EXE and CWSDPMI.EXE should be placed on same directory), then enter the values from 0~255. The system will reboot automatically according to the time-out you set.

```
////////////////////////////////////
```

You can write your own program by modifying the source code F81865_Test.cpp..The index address is 2EH.

```
////////////////////////////////////  
////
```

```
//F81865_Test.cpp:F81865_test.exe utility for F81865.lib  
APIs demonstration.
```

```
//
```

```
// History:
```

```
//      7/15/2011      Brand new F81865_test  
program.
```

```
#include <winsock2.h>
```

```
#include "Windows.h"
```

```
#include "stdio.h"
```

```
#include "F81865.h"
```

```
#define PARAMETER_HELP      "\n\  
    \"The F81865 GPIO utility of Lanner\n\  
    \"----- \n\  
    \"Usage:\n\  
    \" F81865_test DIO_IN      port_  
number\n\  
    \" F81865_test DIO_OUT     port_  
number value\n\  
    \" F81865_test PIO         port_number  
value\n\  
    \" F81865_test RunLED      port_  
number value\n\  
    \" F81865_test AlarmLED    port_  
number value\n\  
    \" F81865_test GPS_LED     port_  
number value\n\  
    \" F81865_test WirelessLED port_  
number value\n\  
    \" F81865_test WatchDog    seconds\n\  
    \" F81865_test CaseOpen\n\  
    \" F81865_test CaseOpen_Clear\n\  
    \" F81865_test Sleep      milliseconds\n\  
    \"\n\  
\"Argement:\n\  
    \" DIO_IN      Read state from DIO  
In.\n\  
    \" DIO_OUT     Set DIO Out state.\n\  
    \" PIO         Set PIO LED state.\n\  
    \" RunLED      Set RUN LED state.\n\  
    \" AlarmLED     Set Alarm LED state.\n\  
    \" GPS_LED      Set GPS LED state.\n\  
    \" WirelessLED  Set Wireless LED state.\n
```

```

        " Watchdog
Watchdog timer.\n"

        " CaseOpen      Check case opened state.\n"

        " CaseOpen_Clear  Clear case open state.\n"
            " port_number    The port number.\n"
            " value          1 for on and 0 for off.\n"
            " seconds        The
watchdog count down seconds. 0 for disable.\n"
            " milliseconds    Milliseconds to
delay\n"

#define RETMSG(a,b) {printf (b) ; return a;}

#define CHECK_ARGC(a) {if (argc != a) throw
PARAMETER_HELP ;}

//Translate Hex string to a long value
LONG Hex2Long (char *str)
{
    LONG nLong ;

    if (scanf (str, "%x", &nLong) != 1)
        throw "Error parsing parameter\n";

    return nLong ;
}

// Make sure the argument is numeric
void CheckNumeric (char *szBuf)
{
    int nLen = strlen (szBuf) ;

    for (int i = 0 ; i < nLen ; i++)
        if (!strchr ("01234567890ABCDEFabcdef", szBuf[i]))
            throw "Wrong argument\n";
}

// Common GPIO output function definition
#define GPIO_OUT(a,b,c)
int a (int argc, char *argv[])
{
    CHECK_ARGC (4) ;

    CheckNumeric (argv[2]) ;
    CheckNumeric (argv[3]) ;
}

Set

    \
    int nPort = atoi (argv[2]) ;\
    int nValue = atoi (argv[3]) ; \

    \
    c (nPort, nValue) ;

    \
    printf (b " #%d = %d\n", nPort, nValue) ; \

    \
    return 0 ;
}

// Function generate by common function definition
GPIO_OUT (mDIO_OUT , "DIO_OUT"
, Write_DIO)
GPIO_OUT (mPIO , "DIO_OUT"
, PIO)
GPIO_OUT (mRunLED , "RunLED"
, RunLED)
GPIO_OUT (mAlarmLED , "AlarmLED"
, AlarmLED)
GPIO_OUT (mGPS_LED , "GPS_LED"
, GPS_LED)
GPIO_OUT (mWirelessLED , "WirelessLED"
, WirelessLED)

// Check case open
int mCaseOpen (int argc, char* argv[])
{
    CHECK_ARGC (2) ;

    BOOL bOpen = CaseOpen () ;
    printf ("Case is %s\n", bOpen ? "Open" : "Close") ;

    return bOpen ;
}

```

```

// Clear case open state
int mCaseOpen_Clear (int argc, char* argv[])
{
    CHECK_ARGC (2);

    CaseOpen_Clear ();

    BOOL bOpen = CaseOpen ();

    printf("CaseOpen state %s", bOpen ? "not
cleared" : "cleared");

    return bOpen ;
}

```

```

// Get DIO_IN state
int mDIO_IN (int argc, char* argv[])
{
    CHECK_ARGC (3);
    CheckNumeric (argv[2]);

    int nPort = atoi (argv[2]);
    BOOL ret = Read_DIO (nPort);

    printf("DIO_IN #%d = %d\n", nPort, ret);

    return ret ;
}

```

```

// Milli-second delay
int mSleep (int argc, char *argv[])
{
    CHECK_ARGC (3);

    CheckNumeric (argv[2]);

    Sleep (atoi (argv[2]) );
}

```

```

return 0 ;
}

// Watchdog
int mWatchDog (int argc, char *argv[])
{
    if (argc != 3 && argc != 2)
        RETMSG (-1, PARAMETER_HELP);

    if (argc == 3)
    {
        CheckNumeric (argv[2]);

        int nValue = atoi (argv[2]) ;

        WatchDog_Enable (nValue);
    }

    int nLeft = WatchDog_GetLeft ();

    printf ("Watchdog timer left %d seconds\n",
nLeft);

    return nLeft ;
}

```

```

// Argument - function mapping
typedef struct
{
    char *szCmd ;
    int (*function) (int argc, char *argv[]) ;
} CMD2FUN ;

CMD2FUN c2f[] =
{

```



```

        {"DIO_IN"          , mDIO_IN
},
        {"DIO_OUT"        , mDIO_OUT
},
        {"PIO"            , mPIO
},
        {"RunLED"         , mRunLED
},
        {"AlarmLED"       , mAlarmLED
},
        {"GPS_LED"        , mGPS_LED
},
        {"WirelessLED"    , mWirelessLED },
        {"CaseOpen"       , mCaseOpen  },
        {"CaseOpen_Clear", mCaseOpen_Clear},
        {"Watchdog"       , mWatchDog
},
        {"Sleep"          , mSleep      }
};

// No match argument
RETMSG (-1, "Wrong Argument\n");
}

catch (char *str)
{
    // Output the error message
    printf ("\n%s\n", str);
}

catch (...)
{
    // Unknown exception
    printf ("\nUnknown Exception\n");
}

return -1;
}

```

```

// Program start here
int main(int argc, char *argv[])
{
    try
    {
        //The total argument allowed
        int num = sizeof (c2f) / sizeof (c2f[0]);

        //Too few argument
        if (argc < 2)
            RETMSG (-1, PARAMETER_
HELP);

        // Find the match argument and
        execute the mapping function
        for (int i = 0 ; i < num ; i++)
            if (strcmp (argv[1], c2f[i].
szCmd) == 0)
                return c2f[i].function
(argc, argv);
    }
}

```

Appendix E:

Terms and Conditions

Warranty Policy

1. All products are under warranty against defects in materials and workmanship for a period of one year from the date of purchase.
2. The buyer will bear the return freight charges for goods returned for repair within the warranty period; whereas the manufacturer will bear the after-service freight charges for goods returned to the user.
3. The buyer will pay for repair (for replaced components plus service time) and transportation charges (both ways) for items after the expiration of the warranty period.
4. If the RMA Service Request Form does not meet the stated requirement as listed on "RMA Service," RMA goods will be returned at customer's expense.
5. The following conditions are excluded from this warranty:

Improper or inadequate maintenance by the customer
Unauthorized modification, misuse, or reversed engineering of the product
Operation outside of the environmental specifications for the product.

RMA Service

Requesting a RMA#

6. To obtain a RMA number, simply fill out and fax the "RMA Request Form" to your supplier.
7. The customer is required to fill out the problem code as listed. If your problem is not among the codes listed, please write the symptom description in the remarks box.
8. Ship the defective unit(s) on freight prepaid terms. Use the original packing materials when possible.
9. Mark the RMA# clearly on the box.



Note: Customer is responsible for shipping damage(s) resulting from inadequate/loose packing of the defective unit(s). All RMA# are valid for 30 days only; RMA goods received after the effective RMA# period will be rejected.

RMA Service Request Form

When requesting RMA service, please fill out the following form. Without this form enclosed, your RMA cannot be processed.

RMA No:		Reasons to Return: <input type="checkbox"/> Repair(Please include failure details)	
		<input type="checkbox"/> Testing Purpose	
Company:		Contact Person:	
Phone No.:		Purchased Date:	
Fax No.:		Applied Date:	
Return Shipping Address: _____			
Shipping by: <input type="checkbox"/> Air Freight <input type="checkbox"/> Sea <input type="checkbox"/> Express _____			
<input type="checkbox"/> Others: _____			
Item	Model Name	Serial Number	Configuration

Item	Problem Code	Failure Status

*Problem Code:

01: D.O.A.

02: Second Time

R.M.A.

03: CMOS Data Lost

04: FDC Fail

05: HDC Fail

06: Bad Slot

07: BIOS Problem

08: Keyboard Controller Fail

09: Cache RMA Problem

10: Memory Socket Bad

11: Hang Up Software

12: Out Look Damage

13: SCSI

14: LPT Port

15: PS2

16: LAN

17: COM Port

18: Watchdog Timer

19: DIO

20: Buzzer

21: Shut Down

22: Panel Fail

23: CRT Fail

24: Others (Pls specify)

Request Party

Confirmed By Supplier

Authorized Signature / Date

Authorized Signature / Date